

Using meta-ethnography to synthesize research on knowledge management and agile software development methodology

Glauco Antonio Ruiz
Bianca Minetto Napoleão
Erica Ferreira de Souza
Department of Computer, Federal
Technological University – UTFPR
Cornélio Procópio, PR, Brazil
{glaucoruiz,biancanapoleao}@
alunos.utfpr.edu.br
ericasouza@utfpr.edu.br

Katia Romero Felizardo
Giovani Volnei Meinerz
Patrick Rodrigo da Silva
Department of Computer, Federal
Technological University – UTFPR
Cornélio Procópio, PR, Brazil
katiascannavino@utfpr.edu.br
giovaniminerz@utfpr.edu.br
patricksap@hotmail.com

Nandamudi L. Vijaykumar
Laboratory of Computing and
Applied Mathematics
National Institute for Space
Research – INPE
São José dos Campos, SP, Brazil
vijay.nl@inpe.br

ABSTRACT

Context: Software development processes are considered as knowledge intensive and therefore Knowledge Management (KM) can be applied to efficiently manage the knowledge generated. Agile practices can benefit the software organizations in terms of KM. Some studies have already presented evidence about this relationship. However, the intersection of these two areas still require further more clarification. **Objective:** This study aims to synthesize research on KM and Agile Software Development (ASD) using the meta-ethnography method. **Method:** In order to achieve the proposed goal, first, we applied the seven phases of meta-ethnography analysis method on a five articles selected from a tertiary study on KM and ASD. Second, the relations identified between the areas investigated were analysed from interviews with three agile development methodology experts. **Results:** A relation map that summarizes the synthesis identified between KM, agile values and scrum activities was created. **Conclusion:** There is a significant contribution in KM and ASD for both software engineering academics and industry.

CCS CONCEPTS

• **Software and its engineering** → **Agile software development**;

KEYWORDS

Software Engineering, Knowledge Management, Agile Software Development

ACM Reference Format:

Glauco Antonio Ruiz Bianca Minetto Napoleão Erica Ferreira de Souza, Katia Romero Felizardo Giovani Volnei Meinerz Patrick Rodrigo da Silva, and Nandamudi L. Vijaykumar. 2018. Using meta-ethnography to synthesize research on knowledge management and agile software development methodology. In *17th Brazilian Symposium on Software Quality (SBQS), October 17–19, 2018, Curitiba, Brazil*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3275245.3275270>

1 INTRODUCTION

A major Software Engineering challenge is the creation of systematic strategies to integrate knowledge involved in a project. Software organizations look for solutions that emphasize different knowledge types when planning initiatives to manage them [3]. A systematic routine for capturing knowledge is important in an organization, making knowledge transparent to all those involved. Without it an organization can not benefit from its accumulated knowledge. However, while the importance of working with the generated knowledge has been widely recognized in many areas, managing that knowledge is still a challenge for software organizations. The principles of Knowledge Management (KM) are mechanisms that can solve, at least partially, such challenges [10].

KM promotes the storage and sharing of knowledge, as well as making organizational knowledge accessible and reusable in the organization [21]. According to O’Leary and Studer [21], KM formally manages knowledge resources in order to facilitate access and reuse. Nonaka and Krogh [20] use the tacit-explicit distinction to differentiate unarticulated and articulated stocks of knowledge. Tacit knowledge is the experience-based knowledge that cannot be documented, and typically remains only in people’s minds. Tacit knowledge covers knowledge that is associated to the senses, skills, experiences or intuition. Explicit knowledge, in turn, represents knowledge that can be documented since it is objective and rational. Explicit knowledge can be easily used and shared.

Traditional software development involves the use of several documents to capture and represent knowledge related to the various stages of software development life cycles [29]. Explicit knowledge is decisive. KM for software engineering aims at facilitating knowledge flow and utilization across

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SBQS, October 17–19, 2018, Curitiba, Brazil
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6565-9/18/10...\$15.00
<https://doi.org/10.1145/3275245.3275270>

every phase of a software engineering process, simplifying the process of creating, capturing and sharing the company knowledge [28].

Unlike traditional software methods, Agile Software Development (ASD) is a lightweight software development. Agile methods emphasize social interactions and collaboration among team members in applying and sharing knowledge. Agile methods emphasize tacit knowledge, encouraging on individual, team and customer communications and interactions [1]. Therefore, agile practices and KM present common activities that can benefit the software organizations to promote knowledge sharing, team communication, knowledge reuse, and collaborative process. Nevertheless, according to Cabral et al. [6], there is still a gap about what emerges from the intersection of these two areas that require further more clarification. For this reason, KM in ASD has been treated as a broad research topic resulting in several relations between its schools and concepts.

Based on the above context, we intend to understand how ASD are inherent to KM activities. We decided to explore the following research question to guide us to evidence the relations in both these areas:

How do agile values and practices relate to knowledge management activities?

In order to answer this research question, we synthesized research on KM and ASD methodology using the meta-ethnography method. Meta-ethnography is an interpretive method to research synthesis with the intention of achieving synthesis by absorbing the concepts identified in the relevant studies into a higher-order theoretical structure [8, 19]. Noblit and Hare [19] emphasized that a meta-ethnography should not attempt to produce gross generalizations across studies with disparate goals and from too distinct contexts. In this sense, it is unlike aggregative methods, such as meta-analysis or integrative reviews, which look for developing synthesis of an exhaustive list of studies, attempting to increase results external validity. First, we applied the seven phases of meta-ethnography analysis method on five articles selected from a tertiary study on KM and ASD. Second, the relations identified between the areas investigated were validated based on interviews with three ASD professionals.

The remainder of this study is structured as follows. Section 2 review the literature of KM and agile development. Section 3 formally introduces the methods and procedures used to conduct the research, as well as the main results. Section 4 reports a general discussion to highlight some points of our research. Lastly, conclusions remarks and future directions are described in Section 5.

2 BACKGROUND

In this section, the main concepts of this study and related works are discussed briefly.

2.1 Knowledge Management

KM is the process through which organizations generate value from their intellectual assets (tacit and explicit knowledge) [5]. Most often, generating value from such assets involves sharing them among employees, departments and even with other companies in an effort to devise best practices.

Effective KM requires an organization to execute activities such as to identify, generate, acquire, diffuse, and capture knowledge. KM cycles, also known as KM models, can be used to guide how to conduct such activities. KM cycles have an objective to help identifying and locating knowledge and knowledge sources within the organization. According to Dalkir [9], the four most known KM cycles are presented by Wiig [30], Meyer and Zack [17], McElroy [15] and Bukowitz and Williams [5]. In summary, these KM cycles identify and locate knowledge and knowledge sources within the organization from several activities. However, Dalkir [9] also mentions that a similar lack of consensus exists with respect to the terms used to describe the major activities in the KM cycle. So, on the basis of main KM cycles, [9] one can distill an integrated KM cycle represented by interaction of three activities:

- **Knowledge capture and/or creation:** Knowledge capture refers to the identification of existing internal and/or external knowledge from the environment. Knowledge creation is the development of new knowledge that did not have a previous existence within the organization. In this activity the tacit knowledge is captured or elicited, and explicit knowledge is organized or coded.
- **Knowledge sharing and dissemination:** Once knowledge has been captured and coded, it needs to be shared and disseminated throughout the organization. There are several practices that can be used to share knowledge within the organization, such as team meetings, written instructions, ad hoc, verbally, intranet or video.
- **Knowledge acquisition and application:** Knowledge that has been captured, coded, shared, and otherwise made available is put to use. KM can succeed only if the knowledge is used. However, it now becomes imperative to understand which knowledge is of use to which set of people and how best to make it available. The use of KM systems, such as expertise location systems or content management systems can be designed to optimize knowledge application on an organization-wide basis.

The transition between the three activities is represented in Figure 1. In integrated KM cycle, the transition from knowledge capture/creation to knowledge sharing and dissemination, knowledge content is assessed. Knowledge is then contextualized in order to be understood (“acquisition”) and used (“application”). This process then feeds back into the first activity in order to update the knowledge content [9].

In this study, the integrated KM cycle presented in [9], will be used to represent the main KM activities serving as input for the synthesis method presented in Section 3.

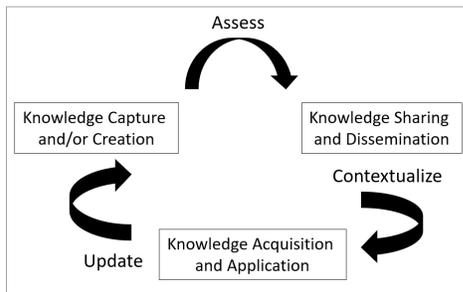


Figure 1: An integrated KM Cycle

2.2 Knowledge Management and Software Engineering

One of the characteristics of software engineering is the high volume of information that is generated and manipulated in the organization. Those involved in the project face problems such as: difficulty in systematizing the information generated throughout the software processes; difficulty in reusing knowledge generated from one project in another; loss of intellectual capital of the organization; and the non-representation of knowledge [27]. From these problems, integrating KM in software engineering has brought much discussion about how to manage knowledge in the organization.

Knowledge in a software engineering organization should be properly captured, stored and reused when needed. KM principles and techniques application aims at facilitating knowledge flow and utilization across software engineering process. In this environment knowledge needs to be updated all the time since software development environment technologies used are often changing [28]. According to Bjørnson and Dingsøyr [3], research related which KM and software engineering focuses on ways and means to share knowledge.

2.3 Agile Software Development

Traditional software development process can limit developers, since it sometimes turns into a complex and expensive process. Besides, it is extremely focused on documentation. This fact highlights the emergence of agile methods [23]. Projects using agile methodologies assume that change is common in software projects (and software-heavy projects) and thus value ongoing planning, emphasizing human aspects and adaptability to rapid changes in the project.

In 2001, leaders of different streams joined together and created the Manifesto for Agile Software Development (ASD) [2]. The Agile Manifesto includes values and principles that help to optimize the software development process and also have a strong influence on present practices of team collaboration within ASD [23]. The Agile Manifesto provides four core values:

- (I) Individuals and interactions over processes and tools;
- (II) Working software over comprehensive documentation;
- (III) Customer collaboration over contract negotiation; and
- (IV) Responding to change following a plan.

Agile values promote a focus on the people involved in a project and how they interact and communicate. In particular, the communication is strongly related to KM, and in agile teams the focus is the tacit knowledge.

There are many methodologies and practices in ASD, and among the best-known and most commonly used are Scrum and Extreme Programming (XP) [6]. In Scrum, the projects are divided into *sprints*, which typically last for few weeks. After each *sprint*, actors and staff members meet to assess project progress and plan their next steps. Similarly to Scrum, XP also has short development cycles. XP includes programming in pairs doing extensive code review, unit testing of entire code, constantly communicating with their customers and fellow programmers, deliver the system to the customers as early as possible and implement changes as suggested.

In order to represent the relation of agile practices with KM, in this study we use Scrum framework as input to the synthesis method. We chose Scrum as an input for this work since it is currently considered the most commonly used method/framework for software development [7].

Scrum Guide [24] official document defines Scrum as a framework for developing, delivering, and sustaining complex products. The guide focuses on two core elements: Scrum events (more precisely in Sprint event, which is a container for all other events) and Scrum artifacts. Scrum events promote meetings focused on critical transparency and inspection reducing several meetings and creating uniformity. Scrum artifacts differ from common artifacts because they are designed to provide transparency and possibility for inspection and adaptation ensuring that everyone has the same understanding about a particular artifact [24].

2.4 Meta-ethnography Method

Noblit and Hare [19] define meta-ethnography as an intensive synthesis method involving observation, interviewing and document review. It enables a systematic and detailed understanding of how studies are related. This relation is done through the comparison of findings within and across studies. Meta-ethnography is the method of qualitative synthesis most widely used in health and education research. According to Kitchenham et al. [13], there are three examples of software engineering systematic reviews that have employed meta-ethnography in order to synthesize the studies [11, 12, 25].

The meta-ethnography method is composed by seven stages, briefly described following:

- (1) Getting started – the starting point involves identifying a topic that qualitative research might inform. It is relevant to find a topic that could be of interest to both researchers and also for practitioners. Researchers can define a research question to represent the topic and guide the meta-ethnography application.
- (2) Deciding what studies are relevant to the topic of interest – the goal of the second stage is to find and select studies that are relevant to the topic of interest. It involves activities such as: searching candidate studies;

making decisions on inclusion/exclusion; and quality assessment.

- (3) Reading the studies – during this stage researchers carefully read the set of included studies becoming as familiar as possible with their general content and detail. Researchers can also incorporate data extraction.
- (4) Determining how the studies are related – initially, researchers can list the key metaphors of each study, which be phrases, ideas and/or concepts. Next, the researchers can create a list of these metaphors, juxtaposing them and determining how they are related. Grids or tables can also be used to display metaphors across all studies.
- (5) Translating the studies into one another – in this stage the goal is to describe how to compare metaphors in one study with those in another. The original method does not define exactly how to do this. One suggestion is to compare the metaphors from study 1 with study 2, and the synthesis of these two studies with study 3, and so on [25].
- (6) Synthesizing the translations – when the previous stage results in many metaphors, these can be compared to see if there are common types or if some metaphors can encompass others. In addition, the findings of the synthesis can be represented as diagrams or figures.
- (7) Expressing the synthesis – the findings of the synthesis can be finally disseminated to interested parties.

In summary, researchers select, analyze and interpret qualitative studies through a process of translation, in which the studies are coded into metaphors, which provide an interpretation of the entire topic, in order to answer focused questions on a specific topic and gain new insights.

3 APPLICATION OF META-ETHNOGRAPHY METHOD

The seven stages suggested by Noblit and Hare [19] were used to construct the meta-ethnography.

(1) *Getting started*

In this study, we intend to understand how agile values and practices are inherent to KM activities. We want to perform a synthesis about the intersection of these two areas that needed further clarification for researchers and practitioners interested in the study of ASD and KM. In regard to ASD, we focus only on Scrum framework. Therefore, we decided to refine and explore the following research question to guide us to evidence the relations in both these areas: *How do agile values and Scrum events and artifacts relate to KM activities?*

(2) *Deciding what is relevant to the initial interest*

In order to decide which studies are relevant to the initial interest we conducted a tertiary study looking for secondary studies investigating the state of the art in KM and agile development. Tertiary studies are considered as a review that focuses only on secondary studies, i.e., it is a review

about other secondary studies [14]. The tertiary study was conducted until December 2017 and synthesized in 2018 by three researchers involved in all phases. We used the search string shown in Table 1. The string was applied in three metadata fields: title, abstract and keywords. The search string had syntactic adaptations according to particularities of each source.

The source used were in this review were: Scopus, IEEE Xplore Digital Library, Science Direct and Engineering Index Compendex. The selection criteria are organized in one inclusion criterion (IC) and five exclusion criteria (EC). The inclusion criterion is: (IC1) The study discusses KM and agile software development methodology. The exclusion criteria are: (EC1) The study is just published as an abstract; (EC2) The study is not written in English; (EC3) The study is an older version of other study already considered; (EC4) The study is not a secondary study, such as primary study or editorials, summaries of keynotes, workshops, and tutorials; and (EC5) The study is not available.

A total of 161 studies were identified during the search process. First, we eliminated duplications (publications that appear in more than one source), achieving 155 publications. Out of these studies, we selected secondary studies by reading their titles and abstracts and applying the inclusion and exclusion criteria. As a result, a total of 19 studies were selected. Next, the selection criteria were applied considering the full text. A total of five studies were returned in this stage. Over these five studies considered relevant, we performed backward snowballing. Snowballing was applied in order to identify additional relevant studies through the reference lists of the five studies. However, no relevant study was identified. Thus, we identified only five relevant studies from the four sources that we searched, presented in Table 2.

In Table 2 the article reference, year of publication, how many primary studies were included in each article and the title are shown. It is worth mentioning that although our initial set is five secondary studies, this set increased considering the primary studies returned in each one. The total of 179 primary articles were returned, which were analyzed when necessary.

(3) *Reading the studies*

Firstly, an individual reading was done by each author to identify the relations between “agile values and KM” and “Scrum and KM”. Subsequently, we conducted several meetings in groups to resolve in consensus the divergences. The following is a brief presentation of the initial set of studies.

Analyzing the five selected papers over the years, the intention to map and begin to correlate KM initiatives in ASD is recent, starting basically in 2011, as Table 2 suggests.

Regarding the scope of each study, although all studies investigate KM and ASD, each study has a different purpose. In [22], the purpose is to summarize information visualization techniques used during the design and development steps of the software cycle by agile teams. [22] suggest that visualization techniques are important in ASD because they

Table 1: Keywords of the Search String of the Tertiary Study on CMs in KM and agile development.

Areas	Keywords
Agile	“agile”, “extreme programming”, “xp method”, “scrum”, “crystal”, “dscdm”, “fdd”, “feature driven development”, “lean software development”
KM	“Knowledge Management”, “knowledge sharing”, “knowledge transfer”, “knowledge extraction”, “Knowledge discovery”, “useful knowledge”, “tacit knowledge”, “explicit knowledge”, “knowledge creation”, “knowledge acquisition”, “knowledge retention”, “knowledge evaluation”, “knowledge application”, “organization knowledge”, “knowledge engineering”, “Knowledge representation”)
Review	“Systematic Literature Review”, “Systematic Review”, “Systematic Mapping”, “Mapping Study”, “Mapping Studies”, “Systematic Literature Mapping”, “Literature Review”

Search String: (“agile” OR “extreme programming” OR “xp method” OR “scrum” OR “crystal” OR “dscdm” OR “fdd” OR “feature driven development” OR “lean software development”) AND (“Knowledge Management” OR “knowledge sharing” OR “knowledge transfer” OR “knowledge extraction” OR “Knowledge discovery” OR “useful knowledge” OR “tacit knowledge” OR “explicit knowledge” OR “knowledge creation” OR “knowledge acquisition” OR “knowledge retention” OR “knowledge evaluation” OR “knowledge application” OR “organization knowledge” OR “knowledge engineering” OR “Knowledge representation”) AND (“Systematic Literature Review” OR “Systematic Review” OR “Systematic Mapping” OR “Mapping Study” OR “Mapping Studies” OR “Systematic Literature Mapping” OR “Literature Review”)

Table 2: Initial set of selected studies

Ref.	Year	N ^o Primary Study	Title study
[1]	2017	48	Understanding Knowledge Management in Agile Software Development Practice
[4]	2017	42	Review of approaches to manage architectural knowledge in Agile Global Software Development
[6]	2014	25	Knowledge Management in Agile Software Projects: A Systematic Review
[18]	2011	14	Knowledge creation and sharing in software development teams using Agile methodologies: key insights affecting their adoption
[22]	2014	50	Information Visualization for Agile Software Development Teams

lead to understanding, collaboration, and self-organization. Visualization techniques help Agile teams to increase knowledge sharing when designing, developing, communicating, and tracking progress.

In [6], was investigated the major KM concepts and findings in ASD projects. The study focuses mainly on main outstanding issues in the software development cycle when tacit knowledge use is prioritized over explicit knowledge. The authors discussed the main findings from the perspective of a paradigm shift that prioritizes the use of tacit knowledge rather than explicit knowledge.

[1] conducted a systematic review looking for specific agile practices to support KM, the inherent knowledge involved in these agile practices and how the agile teams manage that knowledge. According to the authors, the most important contribution the review provided was an understanding of KM in ASD that involves managing three different types of knowledge - process, project and product - by implementing three KM strategies - discussions, artifacts and visualizations - during agile practices.

The main purpose in [4] is architectural KM in Agile and Global Software Development (AGSD). A mapping study was conducted where the authors identified and described approaches to manage architectural knowledge in AGSD teams. These approaches were grouped as documentation artefact-based, communication-based, and methodological-based.

Finally, [18] analyzed and evaluated the knowledge creation and sharing experiences of teams in the ASD domain.

A method was developed to evaluate the advantages and limitations of agile practices in knowledge creation and sharing for Agile teams. Considering SWOT (Strengths, Weaknesses, Opportunities, Threats) analysis, the authors assessed Agile processes, their relationships with knowledge transfer management and their effects on the productivity of software development teams.

Once deeply analyzed each paper we started to identify the key concepts addressed in each individual study, making it possible to determine how these concepts are related. This relationship is shown in the next stage of synthesis method.

(4) *Determining how the studies are related*

In the interest for a better understanding of the concepts found in the selected studies, we intend to answer the following research question in this stage: *What is the study concept presented according to KM and Agile perspectives?* In order to answer this question, we created Table 3 to present the perspectives and their concepts, together with the studies and concepts they have employed.

In relation to KM perspective, knowledge sharing is addressed by all studies. In [1] and [22], for instance, one of the concepts is to promote the use of visualization techniques to improve the sharing and retention of knowledge. Agile principles promote producing less documentation. This fact leads the researchers to explore strategies of information representation in a leaner and more efficient way, as is the case of [1] and [22]. In KM, one of the main problems is how to

Table 3: Concept of the studies according to the KM and Agile perspectives

Perspectives	Concepts	Studies
KM	Sharing	[1],[4], [6], [18], [22]
	Capture/Creation	[18]
	Types	[1], [6]
	Reuse	[6]
	Cycle (all activities)	[1], [6], [22]
	Management Strategies	[1]
Agile	Architectural Knowledge Management	[4]
	Software Artifacts (documentation)	[1], [4], [6], [22]
	Communication Team	[1], [4], [6], [18]
	Collaborative process	[6]
	Agile Practices	[1], [4], [18]
	Architectural Issues	[4]
Technological solutions	[4], [6]	

represent knowledge so that it is easily shared [26]. Visualization techniques can minimize ambiguity and imprecision in interpreting shared information.

In the agile perspective, the two most investigated concepts were Software Artifacts and Communication Team. In [6], for example, one of the study classifications is about problems and specific aspects of project documentation in agile methods. In agile projects face-to-face communication is prioritized, instead of the use of documentation. However, some primary studies returned in [6] suggest that even though the project is agile, if the team is distributed, there should be a greater focus on knowledge sharing through documents.

KM strategies were investigated by [1] (included in our KM perspective). According to [1], with respect to this concept, agile teams use: discussions (e.g. sharing requirements), artifacts (e.g. user stories) and visualizations (e.g. burn-down charts), to manage knowledge. The discussions, in particular, is a verbal communication that involves interaction among agile team members willing to share knowledge. So, in the agile perspective we included the [1]’s work in “Communication Team”.

Other concepts (purposes) and relationships were identified, as presented in Table 3. This stage helped us to determine how the studies are related. Thus, we started to work on mapping the studies into one another, which corresponds to stage five of the synthesizing method, presented below.

(5) *Translating the studies into one another*

Since our research question relates agile values and agile practices, in this stage we present the relation created from stages 3 and 4. During the construction of the relations between KM and ASD, we considered the definition of each KM process activity versus the definition of the agile values and Scrum events and artifacts presented in the literature. In addition, we used the selected secondary studies to support our findings. We focused on the three KM activities described in Section 2.1 and the agile values presented in Section 2.3.

Table 4 illustrates the relation between KM activities and agile values¹.

Table 4: Relation KM x Agile Values

KM activities	Agile Values			
	(I)	(II)	(III)	(IV)
Create/capture and contextualize		*		*
Share, disseminate and assess	*		*	
Acquisition and application		*		*

The study conducted by [18] presents an analysis and evaluation of KM creation and sharing experiences in ASD teams. Through an hermeneutic analysis, the study showed that the values (II) and (III) are directly related to knowledge creation and the values (I) and (III) to knowledge sharing, as illustrates Table 4. In relation to acquisition and application activities, the acquisition of knowledge is the comprehension, amplification and articulation of knowledge in a way that it is internalized. Knowledge application refers to the real use of knowledge that has been captured or created [9]. Working in the software development or to respond to changes, for instance, experience is required. It is only possible to apply certain knowledge after it had be acquired and the main way to acquire knowledge is through experience. These facts correspond to values (III) and (IV) [6].

As mentioned earlier (Section 2.3), in order to demonstrate the relation between agile practices and KM activities, we opted to use only the Scrum framework. Besides the five inicial set articles, we also adopted the Scrum Guide as basis for this stage of our study.

Table 5 presents the relation among each KM activity and Sprint Events & Scrum Artifacts as well as citations from the five secondary studies included in our tertiary study that induce or confirm the respective relation presented. Due to limitation on number of pages in this conference, Table 5, a more detailed information is available online on <https://goo.gl/Y7BfDH>.

¹The agile values will be referenced by their numbers - “I, II, III and IV”, according to the order presented in Section 2.3

Table 5: Scrum Events and Artifacts x KM Activities

SCRUM	Relation between ASD and KM from SCRUM Guide	Specific Citations	General Citations
KM Activity: Create/Capture and contextualize			
Sprint Planning	The work to be performed in Sprint is planned by collaborative work of the entire team. New knowledge can be generated through discussion of the backlog items.	–	“...daily meetings and other activities of interaction among stakeholders are common. Knowledge is potentially stored in people’s memory...” [6] “Physical forms that contain specific product features and project information, for example... Product backlog and Sprint backlog... ...code repositories...” [1]
Daily Scrum	Development team improves communication, eliminate other meetings, identify and remove development impediments, highlight and promote rapid decision making and improve the knowledge level.	“Face-to-face communication in daily meetings (SCRUM), ... [18]. ...in a daily stand-up meeting agile teams clarify their cumulative work done by the team...” [1] [1]	
Development work	During the Development Work, development team members create and build increments.	–	
Product Backlog	Product Backlog is an ordered list of everything that is known to be necessary in the product. It is the unique source of the requirements for any changes to be made to the product.	“... such as product backlog... ...were collectively classified as artifacts since they contained useful knowledge about the software requirements... The product backlog also helped capture product knowledge...” [18]	
Sprint Backlog	Team modifies the Sprint Backlog (plan) during the Sprint. Backlog items are discussed and negotiated with the Product Owner.	–	
KM Activity: Share, disseminate and assess			
Daily Scrum	Team answer question and discuss what was done what will be done and any impediment to achieving the sprint goal.	“...daily meetings with the team (daily scrum) so that knowledge is shared inside the team” [6].	“...agile methodologies, such as SCRUM, improves organizational learning. ...increase tacit knowledge exchange...”. “Verbal communication that involves interaction among agile team members to share knowledge...” [6] ...knowledge was shared during discussions... sprint review, testing, and small releases...” [1]
Sprint Review	Development team discusses what went well, what problems occurred and how these issues were resolved during Sprint.	–	
Sprint Retrospective	It is made an evaluation of how the sprint was conducted in order to identified improvements to next Sprint.	“In a sprint retrospective project knowledge is shared...” [1]	
Development work	Development team implements functionality and technology. So, as Scrum promotes collaboration, during development work developers communicate whenever it is necessary.	“Agile teams to increase knowledge sharing when designing, developing...” [22] “Verbal communication that involves interaction among agile team members which aims to share knowledge...” [1]	
Product Backlog	Product Owner discusses the Product Backlog as it stands, and it is evaluated at each increment delivery.	“Artifacts in agile practices were commonly used to share product knowledge... (e.g. in the form product backlog).” [1]	
Sprint Backlog	The Sprint Backlog is disseminated and discussed during the sprint and at the end of the sprint, it becomes an increment.	–	
Increment	The increment is shared (presented), reviewed and assessed in the Sprint Review.	–	
KM Activity: Acquisition and application			
Development work	New knowledge is acquired and others are reinforced. The development teams have all necessary skills as a team to create the product increment.	–	–
Increment	An Increment is a body of inspectable, done work that supports empiricism at the end of the sprint. The increment is a step towards a vision or a goal.	“Agile methods provide deliverables after each iteration, ..., thereby facilitating interaction, trust and understanding between on-site customers and the developers” [18]	

Analyzing some relationships identified from Table 5, it is possible to affirm that the sharing, dissemination and access activities are included in all Scrum artifacts and in almost all Scrum events. Only in the Sprint Planning is not directly identified by literature considered. However, during this Scrum event there is indirect information sharing and dissemination between the Product Owner and the scrum team when they are discussing the Sprint purpose and the Product Backlog items that if completed, they will reach the Sprint goal [24].

Event related to development is not explored in details in the Scrum Guide [24]. It is mentioned, but no details are reported on the aspects of its practice. During the research and reflection about the relation between the KM activities and Scrum events, we could notice that the development work is present in all three activities of KM (Table 5). This fact is justified by Scrum values that construct software in a collaborative way.

Few Scrum artifacts and events are related to acquisition and application activities within the context of KM activity. We have identified only one event (development work) and one artifact (increment) that are directly related to this KM activity. This fact induces that more research on the acquisition and application KM activity and ASD should be explored.

(6) *Synthesizing translations*

According to meta-ethnography method [19], in stage six, a synthesis of the translations is constructed. Thus, we created a relation map that summarizes the synthesis identified. The relation map is presented in Figure 2.

In Figure 2 we evidenced all the concepts and their relationships identified in this research. Figure 2 presents the synthesis of translations in relation to agile values, events and scrum artifacts and KM activities. Once created this synthesis of translations, we decided to validate it by conducting interviews with domain experts. The validation phase is not part of meta-ethnography method. However, we consider this process of validation important to ensure the validity of the concepts and relations between ASD and KM identified, since the synthesis of translations was created from specific literature, however, we deem it important to have a validation from a practical vision of professionals who work for many years with agile development.

We conducted a semi-structured interview with three ASD professionals. Semi-structured interviews allow for improvisation and exploration of the studied objects [31]. Interviewees were chosen based on the deep experience with ASD and on their availability and willingness to participate in the research. The interviewees have on average 11 years of experience with ASD, focusing on Scrum and XP methods.

The interview questions were created considering each relationship in our synthesis of translations. For each relationship we create a question in the affirmation format, for example, for the relationship between the concepts “**Create/capture and contextualize**” and “**Sprint Planning**” (see Figure 2) the

following statement was created: “*In agile methods, the Sprint Planning event is the time when the team meets to plan and organize the work to be done. It is at that moment that the internal knowledge is identified and created, that is, at this moment new knowledge can arise. In this activity, the group tacit knowledge is captured.*” In order to answer each affirmation, the interviewee chose a degree of agreement considering a scale of 1 to 5 based on the Likert Scale method [16].

For each closed question presented earlier, we have created an open question with the intention of identifying examples or reports of the respondent’s life experience in relation to their response. In this moment, new questions spontaneously have arisen, which allowed for a better conversation flow and better information. All interviews occurred in June 2018 through Skype videoconference and lasted approximately one hour each. Their answers were transcribed for analysis and interpretation. The whole interview with its questions and the answers format can be accessed at <https://goo.gl/js5jJi> (in Portuguese).

In relation to the relationships validation created in 2, 14 were confirmed from the three interviewees answers with scales equaling to 5; this represents 82.3 % of the synthesis of translations relationships. Out of the other three questions in which there was no consensus on full agreement, two interviewees pointed out a conformance on scale 4, which is a strong indication of the relationships created. However, in one of the questions the interviewee answered a scale 2 for agreement. Basically the question stated that in Sprint Backlog new explicit knowledge is generated and organized (“**Create/capture and contextualize**” x “**Sprint Backlog**”). We asked the interviewee why he did not fully agree with the statement, and the answer was that in his opinion the most of the knowledge is generated in the sprint planning process. In sprint planning the tasks are discussed and explained, but in the sprint backlog, although there may be increases in knowledge, it is little relevance.

The agreement of two area experts with the “**Create/capture and contextualize**” x “**Sprint Backlog**” relationship still indicates the relationship existence. Even so, this item will be discussed in the item of threats to validity.

The interviewees also made important positive notes and reports of the lived experiences. In summary, all the interviewees emphasize, in the open questions, that the creation, transfer or retention of knowledge within the companies that work with agile development are of great importance. The experts mentioned that companies want to be agile, but they do not understand the effort in relation to the change that needs to happen. Companies are very much attached to contract, scope and deadline, but what the companies should do is deliver the value (what the customer needs).

(7) *Expressing the synthesis*

In this last stage the objective to report the synthesis result. According to [25], the target audience is the research community interested in performing synthesis of empirical research using meta-ethnography. Thus, we believe that our

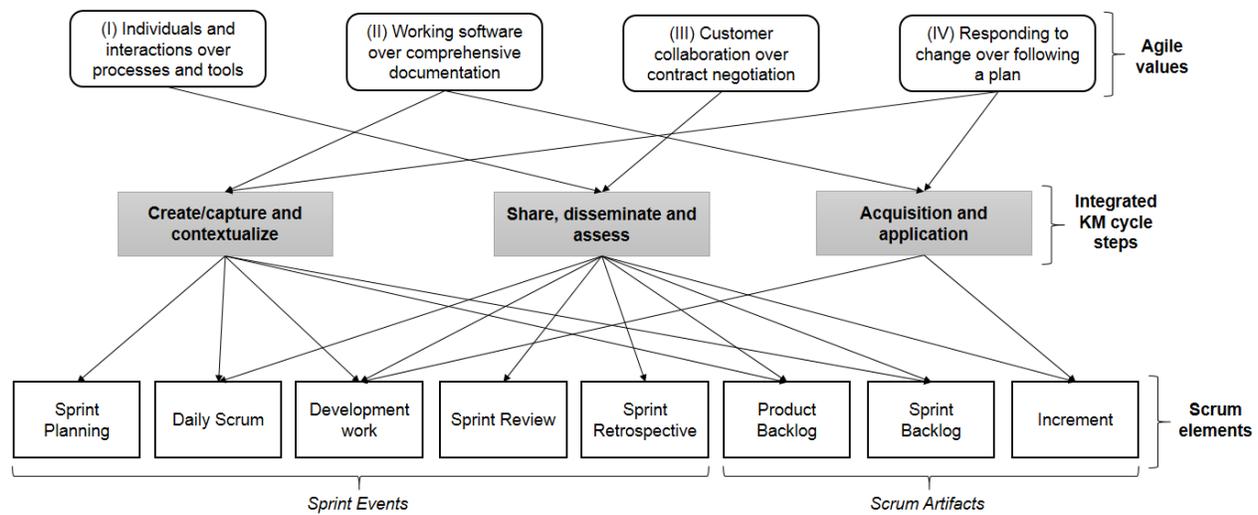


Figure 2: Synthesis of Translations: Relation Map of the relationships between KM, agile values and scrum elements (events and artifacts)

synthesis can help to direct researchers in future research providing a direction to appropriately position new research activities.

4 DISCUSSION

We began this study with the intention of answering the following question “*How do agile values and practices relate to KM activities?*”. Meta-ethnography method was used to identify the existing synthetics between these areas. As pointed out by [25], this method requires several readings of the initial set of studies, data extraction, consensus meeting, reaching agreements and verifying potential inconsistencies in interpretations, time consuming and requires maturity in the research team. However, although the method is not simple to apply, we believe that we have been able to answer the research question by means of a synthesis of translations between KM, agile values and scrum elements (events and artifacts) considered plausible of comprehension. This synthesis of translations created can provide a contribution mainly to researchers and interested in understanding the relationships in this domain: KM and ASD.

A synthesis study focuses primarily on primary studies. However, we realized that the method could also be applied in secondary studies returned from a tertiary study. And even working with five articles in the initial set, when necessary we consulted some of 179 primary studies returned in our five secondary studies.

In relation to the study results, we can say that knowledge capture and sharing are KM activities that are strongly related to ASD and very investigated. This makes sense since ASD prioritizes the exchange of information and communication among teams. Agile values and principles promote a focus on the people and how they interact and communicate. However, we note that the activity of acquisition and application has not been investigated like the other activities.

Knowledge applying activity is the last step of the KM cycle which suggests that the captured and shared knowledge must ultimately be used in the organization. But, the great challenge of software organizations still focus on how to capture and share that knowledge.

The major contribution of this study was to provide the understanding of how specific dimensions of KM in ASD can help area practitioners effectively manage the knowledge in everyday agile practices. Several organizations that adopt agile methodologies face a problem related to the management and organizational knowledge retention. The agile premise is guided by contributory and collaborative work as well as knowledge evolution. The clarification of how KM is present in each agile value, Scrum event and artifacts allows a reflection on how much knowledge has been created, shared and applied during ASD. This reflection enables organizations explore more each KM cycle phase, consequently, contributing to a delivery with greater value to the client.

In addition, the conduction of the meta-ethnography method can also be considered a contribution because it presents another example of how to relate two different areas as well as how to apply the method in practice.

4.1 Threats to Validity

One threat to validity in this study could be the research method used. Since meta-ethnography is an interpretive approach to synthesis, we addressed validity and reliability of our synthesis from an interview with three ASD experts from a practical view.

A larger number of interviews could be consulted and improve this research quality; however, given the interviewees profile, we believe that a single interview could already bring important observations to the research. Even so, we intend to do more interviews and conduct a survey in order to continue validating our synthesis. In the interview that was conducted,

one of the interviewees did not fully agree with a relationship. As mentioned earlier, more interviews will be conducted to minimize possible biases.

Some relationships between KM, agile values and practices may not have appeared in our relation map. This is due to the fact that we considered only five secondary studies and the Scrum guide. An alternative to this limitation, would be to increase the set of references considered during the relationships construction and to consider a greater number of interviewees in the validation process.

5 CONCLUSIONS

In this study, we reported on the results from an analysis on KM and ASD methodology using the meta-ethnography method. We applied the seven phases of meta-ethnography analysis method on five articles selected from a tertiary study on KM and ASD. After, the syntheses identified in these areas investigated were analyzed based on interviews with three ASD consultants.

Agile practices and KM present common activities that can encourage software organizations to promote KM activities. The exploration of KM activities during ASD can improve team learning and collaborate with the evolution of organizational knowledge leading to deliveries with greater value and consequently increase the customer satisfaction. The most common activities between these two areas are knowledge creation and sharing. In [6], they presented that there is still a gap about what emerges from the intersection of these two areas that needs further clarification. In our investigation, we believe that the acquisition and application activities are research topics that can be explored more.

A richer investigation with better mechanisms to perceive KM in ASD forms part of future work. We intend to complement our research with other empirical methods, for example, we intend to conduct a survey in software organizations and to extend this analysis adopting others agile methodologies such as XP.

ACKNOWLEDGMENT

The authors would like to acknowledge the ASD experts that agree to participate in this research.

REFERENCES

- [1] Y. Andriyani, R. Hoda, and R. Amor. 2017. Understanding Knowledge Management in Agile Software Development Practice. International Conference on Knowledge Science, Engineering and Management (KSEM), Melbourne, VIC, Australia, 195–207.
- [2] K. Beck, M. Beedle, van Bennekum A., A. Cockburn, W. Cunningham, M. Fowler, and et al. 2001. Manifesto for Agile Software Development. “<http://www.agilemanifesto.org/>”.
- [3] F. O. Bjørnson and T. Dingsøyr. 2008. Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Information and Software Technology* 50 (2008), 1055–1068.
- [4] G. Borrego, A. L. Morán, R. R. Palacio Cinco, O.M. Rodríguez-Elias, and E. García-Canseco. 2017. Review of approaches to manage architectural knowledge in Agile Global Software Development. *IET Software* 11, 3 (2017), 77–88.
- [5] W. Bukowitz and R. L. Williams. 2000. *The knowledge management fieldbook*. Financial Times Prentice Hall, Great Britain.
- [6] A.R.Y. Cabral, M.B. Ribeiro, and R.P. Noll. 2014. Knowledge Management in Agile Software Projects: A Systematic Review. *Journal of Information & Knowledge Management* 13 (2014), 1–37.
- [7] C.G. Cobb. 2015. *The Project Manager’s Guide to Mastering Agile: Principles and Practices for an Adaptive Approach*. Wiley.
- [8] D. S. Cruzes and T. Dybå. 2008. Research synthesis in software engineering: tertiary study. *Information and Software Technology* 53 (2008), 440–455.
- [9] K. Dalkir. 2005. *Knowledge Management in Theory and Practice*. Elsevier, USA.
- [10] T. H. Davenport and L. Prusak. 2000. *Working knowledge* (2 ed.). Harvard Business School Press, Boston, USA.
- [11] T. Dyba and T. Dingsøyr. 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology* 50 (2008), 833–859.
- [12] Q. Gu and P. Lago. 2009. Exploring service-oriented system engineering challenges: a systematic literature review. *Service Oriented Computing and Applications* 3 (2009), 171–188.
- [13] B.A. Kitchenham, D. Budgen, and P. Brereton. 2015. *Evidence-Based Software Engineering and Systematic Reviews*. Chapman and Hall/CRC.
- [14] B.A. Kitchenham and S. Charters. 2007. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Technical Report EBSE 2007-001. Keele University and Durham University, UK.
- [15] M. Levy and O. Hazzan. [n. d.]. The knowledge life cycle. In *ICM Conference on KM*. Miami, FL.
- [16] Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology* (1932).
- [17] M. Meyer and M. Zack. 1996. The design and implementation of information products. *Sage Publications Inc* (1996).
- [18] F. T. Neves, V. N. Rosa, A. M. R. Correia, and M. d. C. Neto. 2011. Knowledge creation and sharing in software development teams using Agile methodologies: Key insights affecting their adoption. In *6th Iberian Conference on Information Systems and Technologies (CISTI)*. 1–6.
- [19] G.W. Noblit and R.D. Hare. 1988. *Meta-Ethnography*. Sage Publications Inc, Newbury Park, Calif.
- [20] I. Nonaka and G. Krogh. 2009. Tacit Knowledge and Knowledge Conversion: controversy and Advancement in Organizational Knowledge Creation Theory. *Organization Science* 30 (2009), 635–652.
- [21] D.E. O’Leary and R. Studer. 2001. Knowledge Management: an Interdisciplinary Approach. *IEEE Intelligent Systems* 16 (2001).
- [22] J. Paredes, C. Anslow, and F. Maurer. 2014. Information Visualization for Agile Software Development Teams. *IEEE Working Conference on Software Visualization*, 157–166.
- [23] E. M. Schön, M. Escalona, and J. Thomaschewski. 2015. Agile Values and Their Implementation in Practice. *Journal of Interactive Multimedia and Artificial Intelligence* (2015).
- [24] K. Schwaber and J. Sutherland. 2017. *The Scrum Guide*. <https://www.scrumguides.org/download.html>.
- [25] F.Q.B. Silva, S.S.J.O. Cruz, T.B. Gouveia, and L.F. Capretz. 2013. Using Meta-ethnography to Synthesize Research: A Worked Example of the Relations between Personality and Software Team Processes. In *7th International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 153–162.
- [26] E. F. Souza. 2014. *Knowledge management applied to software testing: an ontology based framework*. Ph.D. Dissertation. Instituto Nacional de Pesquisas Espaciais (INPE), Brazil.
- [27] E. F. Souza, R. A. Falbo, and N. L. Vijaykumar. 2015. Knowledge management initiatives in software testing: A mapping study. *Information and Software Technology* 57 (2015), 378–391.
- [28] S. Vasanthapriyan, J. Tian, and J. Xiang. 2015. A Survey on Knowledge Management in Software Engineering. International Conference on Software Quality, Reliability and Security Companion (QRS-C), Vancouver, BC, Canada, 237–244.
- [29] P. Wendorff and D. Apsvalka. 1998. The Knowledge Management Strategy of Agile Software Development. 6th European Conference on Knowledge Management, Univ. of Limerick, Ireland, 607–614.
- [30] K. Wiig. 1997. *Knowledge management foundations*. Schema Press, Arlington, TX.
- [31] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.